306. In yet another embodiment, the signatory 500 inputs the private and public key pair 202, 204 (respectively) through the input device 312.

[0042] The key-verification means 404 verifies 504 the authenticity of the accessed private and public key pair 202, 204 (respectively). As depicted in Figure 2, the signing program 400, which contains the key-verification means 404, could access a verification service 230 via network connection 308. The verification service 230 could be a public key depository, a certificate depository, a certificate or key pair generator, or certificate authority. Verification can be achieved by authenticating the private key. In one embodiment, the key-verification means 404 encrypts a string of data, random or meaningful, using the private key 202 and sends it together with the unencrypted string of data to the verification service 230. The verification service 230 uses the latest published certificate of the signatory 500 to decrypt the encrypted string of data and compares it with the original string. If they match, then the private key 202 is authentic. In another embodiment, the key-verification means 404 obtains the latest certificate of the signatory 500 from the verification service 230 and determines if it matches with the public key 204. If it matches, the private key 202 is authentic. The key-verification means 404 may optionally choose to verify the verification service 230 before trusting the public certificate it returned. Alternatively, the signatory 500 could self-certify and thus provide the verification to the signing program 400. In this embodiment, the signatory 500 is also the issuer of the certificate and the key pair 202, 204, and acts as the verification service 230 to verify the authenticity of the keys to the signing program.

[0043] If the keys 202, 204 are not authentic 506, the signing program 400 alerts the signatory 500 that he can either: (1) retry the process; (2) select or provide a different key pair to the signing program 400; or (3) terminate use of the signing program 400. The key pair 202, 204 may fail to be authenticated for several reasons. For example, the keys may have expired or been revoked. They may have also been mis-entered or otherwise incorrectly supplied by the signatory 500. In any of the foregoing events, if the keys are not valid and/or are not the signatory's keys, the signing program 400 will not use them to generate a digital signature.

[0044] Assuming the keys 202, 204 are properly authenticated, the signing program's 400 universal-signature-object generating means 408 creates a universal signature object by storing 510 a version of the digital data 200. The digital data 200 may be data of any type, such as a text document, an executable file, or any other file. Referring to the example used in connection with the description of the USO 100 in Figure 1, the data may be a business contract generated by Microsoft Word®. The version 102 stored 510 in the USO will have a Microsoft Word® format. The signing program records 512 that the version 112 format is compatible with Microsoft Word®. Alternatively, the signing program 400 searches the computer system 300 on which the signing program 400 operates or searches a network connected to the computer system 300 via a network connection 308 to legally obtain a copy of Word® and include it as part of the information 106 concerning the application.

[0045] The signing program's 400 universal-signature-object generating means 408 prompts the signatory if he would like to store 514 an alternate version 550 of the digital data 200. The signatory can select 530 an existing, but different, version 550A of that data 200 or have an application generate another version of the data 550B. Alternatively, the generating means 408 may automatically produce alternate versions 550 without prompting. In one embodiment, the signing program 400 launches an application that the signatory uses to convert the data 200 into another format. In another embodiment, the signing program includes the ability to convert between multiple file formats. In yet another embodiment, the signatory 500 provides the alternate version 550 or uses an application to create an alternate version 550. Continuing with the business document example, the first version 102 of the business contract was stored as a Microsoft Word® document file. The signatory selects or generates 530 the data 200 in a different format, such as a WordPerfect® format. That version 550 is stored 510 in the USO. Because the signing program has associated at least one application (Microsoft Word®)

compatible with at least one of the version (the first version 102), the step of including 512 information 106 about an application compatible with the version 550 may optionally be excluded. The process of including versions (steps 510, 512, 514, 530) continues until the signatory wishes 514 to include no additional versions of the digital data 200. For the purposes of the continuing business contract illustration, assume the signatory stores a third version 104 of the business contract in a rich text format.

[0046] It is beneficial to have alternate versions of the digital data 200. An alternate version, particularly a version that is compatible with more than one application, such as the third version (rich text format) of the business contract example, increases the value and longevity of the USO 100. More individuals and businesses can access the data 200 and can access it for a longer period of time because there is less reliance on a single, specific format. Furthermore, this portability of the data among multiple applications provides for better archiving. If in the future a person or business needs to verify the digital data 200 (along with a digital signature or signatures), having the data 200 in multiple versions or in a portable/generic format increases the chances that an application can be located to access the data 200. Thus, if an application that generated a version (i.e., the native application), ceases to exist, one of the alternate versions most likely can be utilized.

[0047] It may also be beneficial to have alternate versions if a third party who will utilize the universal signature object 100 may only accept certain formats. Using business contract example, the signatory may use Microsoft Word®, but the party it is contracting with may use only StarOffice™. The parties can utilize the USO as a means for transaction by providing different format versions of the data 200. Each party can utilize the data 200 without incompatibility problems, and each party can include its signature to the agreement (as will be explained in more detail below).

[0048] When the signatory finishes storing versions of the digital data 220, the signing program 400 creates 516 a digital signature. The USO generating means 408 generates 516 a